Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (currently amended) A method of optimizing the operation of ~~operating~~ an n-dimensional array of processing elements ~~to determine a global extrema for said n-dimensional array of processing elements, the method~~ comprising:

determining a local extrema for each of said processing elements;

serially outputting, on each clock cycle, said local extrema from each of said processing elements to a neighboring processing element until every processing element in a first dimension has received all local extrema along said first dimension;

determining within each of said processing elements a first dimensional extrema for said first dimension of said n-dimensional array, wherein said dimensional extrema is determined concurrently with the receipt of ~~from~~ said local extrema from ~~of~~ said processing elements in said first dimension;

serially outputting, on each clock cycle, said first dimensional extrema from each of said processing elements to a neighboring processing element until every processing element in a next dimension has received all first dimensional extrema along said next dimension;

determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array, wherein said next dimensional extrema is determined concurrently with the receipt of ~~from~~ said first dimensional extrema;

repeating said serially outputting on each clock cycle and concurrently determining within each of said processing elements a next dimensional extrema for each of said n-dimensions, wherein each of said next dimensional extrema is determined from a dimensional extrema from a previously selected dimension, until the global extrema is determined; and

saving said global extrema.

2. Cancelled.

3. (previously presented) The method of claim 1 wherein said determining a local extrema comprises:

separating a set of input values within said processing element into an odd set corresponding to values in odd positions within said set of input values and an even set corresponding to values

in even positions within said set of input values, and saving each of said odd set and said even set in first and second registers, respectively;

determining a first extrema from said odd set, and saving said first extrema in a third register;

determining a second extrema from said even set, and saving said second extrema in a fourth register; and

determining said local extrema from said values in said third and fourth registers.

4. (previously presented) The method of claim 1 wherein said determining within each of said processing elements a dimensional extrema for a first dimension of said n-dimensional array comprises:

serially receiving a set of local extrema from said processing elements within said first dimension;

separating said set of local extrema into an odd set corresponding to values in odd positions within said set of local extrema and an even set corresponding to values in even positions within said set of local extrema, and saving each of said odd set and said even set in first and second registers, respectively;

determining a first extrema from said odd set, and saving said first extrema in a third register;

determining a second extrema from said even set, and saving said second extrema in a fourth register; and

determining said dimensional extrema for a first dimension from said values in said third and fourth registers.

5. (previously presented) The method of claim 1 wherein said determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array comprises:

serially receiving a set of said dimensional extrema for a first dimension from said processing elements within said next dimension;

separating said set of dimensional extrema for a first dimension into an odd set corresponding to values in odd positions within said set of dimensional extrema and an even set corresponding to values in even positions within said set of dimensional extrema, and saving each of said odd set and said even set in first and second registers, respectively;

determining a first extrema from said odd set, and saving said first extrema in a third register;

determining a second extrema from said even set, and saving said second extrema in a fourth register; and

determining said next dimensional extrema for a next dimension from said values in said third and fourth registers.

6. (previously presented) The method of claim 1 wherein said repeating said determining within each of said processing elements a next dimensional extrema for each of n-said dimensions comprises:

serially receiving a set of dimensional extrema from said processing elements within a previously selected dimension;

separating said set of dimensional extrema from said previously selected dimension into an odd set corresponding to values in odd positions within said set of dimensional extrema and an even set corresponding to values in even positions within said set of dimensional extrema, and saving each of said odd set and said even set in first and second registers, respectively;

determining a first extrema from said odd set, and saving said first extrema in a third register;

determining a second extrema from said even set, and saving said second extrema in a fourth register; and

determining said next dimensional extrema for said next dimension from said values in said third and fourth registers.

7. (previously presented) The method of claim 4 wherein determining within each of said processing elements a dimensional extrema for a first dimension of said n-dimensional array comprises:

loading a value from an odd position within said set into said first register;

loading a value from an even position within said set into said second register and transferring said value within said first register to said third register;

loading a value from a next odd position within said set into said first register and transferring said value within said second register to said fourth register; and

loading a value from a next even position within said set into said second register.

8. (previously presented) The method of claim 7 wherein said determining a first extrema from said odd set comprises comparing the value in said first register to the value in said third register.

9. (previously presented) The method of claim 8 further comprising:

selecting the greater value from said first register and said third register if a high first extrema is desired; and

selecting the lesser value from said first register and said third register if a low first extrema is desired.

10. (previously presented) The method of claim 7 wherein said determining a second extrema from said even set of values comprises comparing the value in said second register to the value in said fourth register.

11. (previously presented) The method of claim 10 further comprising:

selecting the greater value from said second register and said fourth register if a high second extrema is desired; and

selecting the lesser value from said second register and said fourth register if a low second extrema is desired.

12. (previously presented) The method of claim 8 further comprising:

updating said first extrema in said third register;

loading another value from an odd position within said set into said first register;

comparing the value in said first register to the value in said third register; and

repeating said updating, loading and comparing steps for remaining values within an odd position within said set.

13. (previously presented) The method of claim 10 further comprising:

updating said second extrema in said fourth register;

loading another value from an even position within said set into said second register;

comparing the value in said second register to the value in said fourth register; and

repeating said updating, loading and comparing steps for remaining values within an even position within said set.

14. – 30. Cancelled.

31. (currently amended) An n-dimensional array of processing elements, comprising:

a plurality of processing elements interconnected to form an n-dimensional array, each processing element comprising:

an arithmetic logic unit;

condition logic responsive to said arithmetic logic unit;

a plurality of registers connected to a bus and responsive to said arithmetic logic unit;

a result pipeline responsive to said arithmetic logic unit;

an interface; and

register files connected between said interface and said result pipeline; said processing elements configured to:

determine a local extrema;

serially output, on each clock cycle, said local extrema to a neighboring processing element until every processing element in a first dimension has received all local extrema along said first dimension;

determine a first dimensional extrema for said first dimension of said n-dimensional array, wherein said dimensional extrema is determined concurrently with the receipt of ~~from~~ said local extrema from ~~of~~ said processing elements in said first dimension;

serially output, on each clock cycle, said first dimensional extrema to a neighboring processing element until every processing element in a next dimension has received all first dimensional extrema along said next dimension;

determine a next dimensional extrema for a next dimension of said n-dimensional array, wherein said next dimensional extrema is determined concurrently with the receipt of ~~from~~ said first dimensional extrema;

repeat said serially outputting on each clock cycle and concurrently determining for each of said n-dimensions, wherein each of a next dimensional extrema is determined from a dimensional extrema from a previously selected dimension, until a global extrema is determined; and

save said global extrema.

32. (currently amended) The array of processing elements of claim 31 ~~30~~ wherein said processing elements are configured to:

separate a set of serially received input values into an odd set corresponding to values in odd positions within said set of input values and an even set corresponding to values in even positions within said set of input values, and saving each of said odd set and said even set in first and second registers, respectively;

determine a first extrema from said odd set, and saving said first extrema in a third register;

determine a second extrema from said even set, and saving said second extrema in a fourth register; and

determine an extrema from said values in said third and fourth registers.

33. (currently amended) The array of processing elements of claim 32 ~~31~~ wherein said processing elements are configured to:

select the greater value from said first register and said third register if a high first extrema is desired; and

select the lesser value from said first register and said third register if a low first extrema is desired.

34. (currently amended) The array of processing elements of claim 32 ~~31~~ wherein said processing elements are configured to:

select the greater value from said second register and said fourth register if a high second extrema is desired; and

select the lesser value from said second register and said fourth register if a low second extrema is desired.

35. (currently amended) The array of processing elements of claim 32 ~~31~~ wherein said processing elements are configured to:

select the greater value from said third and fourth registers if a local high extrema is desired; and

select the lesser value from said third and fourth registers if a local low extrema is desired.

36. (currently amended) The array of processing elements of claim 33 ~~32~~ wherein the processing elements are configured to:

serially load another value from an odd position within said set into said first register;

compare the value in said first register to the value in said third register; and

repeat said storing, loading and comparing steps for remaining values within odd positions within said set.

37. (currently amended) The array of processing elements of claim 34 ~~33~~ wherein said processing elements are configured to:

serially load another value from an even position within said set into said second register;

comparing the value in said second register to the value in said fourth register; and

repeat said storing, loading and comparing steps for remaining values within even positions within said set.